CUSTOMER

DSDC

PARTNERS
in Quality Solutions

INDUSTRY

Project Estimation Overview

Presented By: DSDC

sepg@dsdc.dla.mil

CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

Description and Objectives

Description: This course provides an overview of the DSDC Project Estimation Methods which are part of our SEI CMM Levels 2 & 3 effort, and which will increase the probability of a reliable estimate

Objectives:

- **To understand the basic principles of repeatable, consistent estimation of software development projects**
- **To understand the relationship between good requirements and good estimates**

CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

Estimation Goals

- Consistency
- Documentation and History of Estimates
- Continuing effort to bring estimates closer to actuals

PPMT, through its shared database provides the foundation for documenting the entire project history

When Does DSDC Give Estimates?

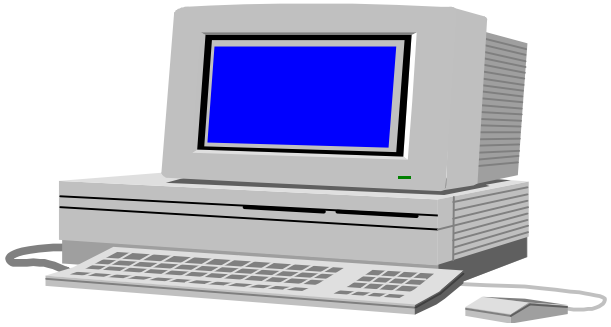
- Preliminary Cost Estimate (Quick Look)
- Proposal Estimate
- Software Development Plan Estimate
- Rebaseline Estimate

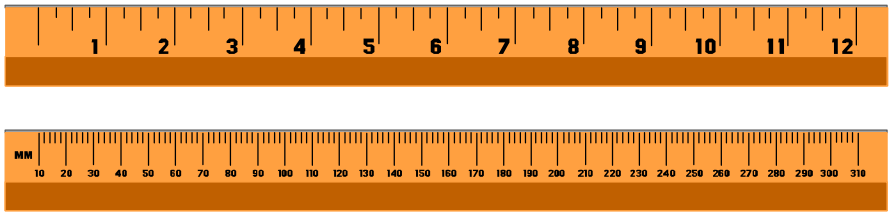
How much will it Cost? $$$$

- How big is it?
  - » Lines of Code
  - » Function Points
  - » Pages of Documentation

COST ESTIMATION

- How much effort will it take?
  - » FP/workhour
  - » COCOMO





Basic Estimating Methods

Professional Judgment

- "My experience and training tell me..."

Analogy

- "This job is similar to another one..."

Automated Tools

- "The model results show that..."

CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

Estimation techniques we have used

- System Experts
- Task breakdown and effort estimate
- Tools (Swan)
- Schedule Driven
- The Hat Method
- “And then Add xx%”

CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

DSDC Estimation Guidelines

- Estimate using at least two techniques
  - Validate models/productivity rates against knowledge, experience, common sense
- Get estimates from independent sources
- Adjust for the people doing the work
- Utilize the Pert Equations - Reduce Risk
- Do it as a team, and document your assumptions and rationale

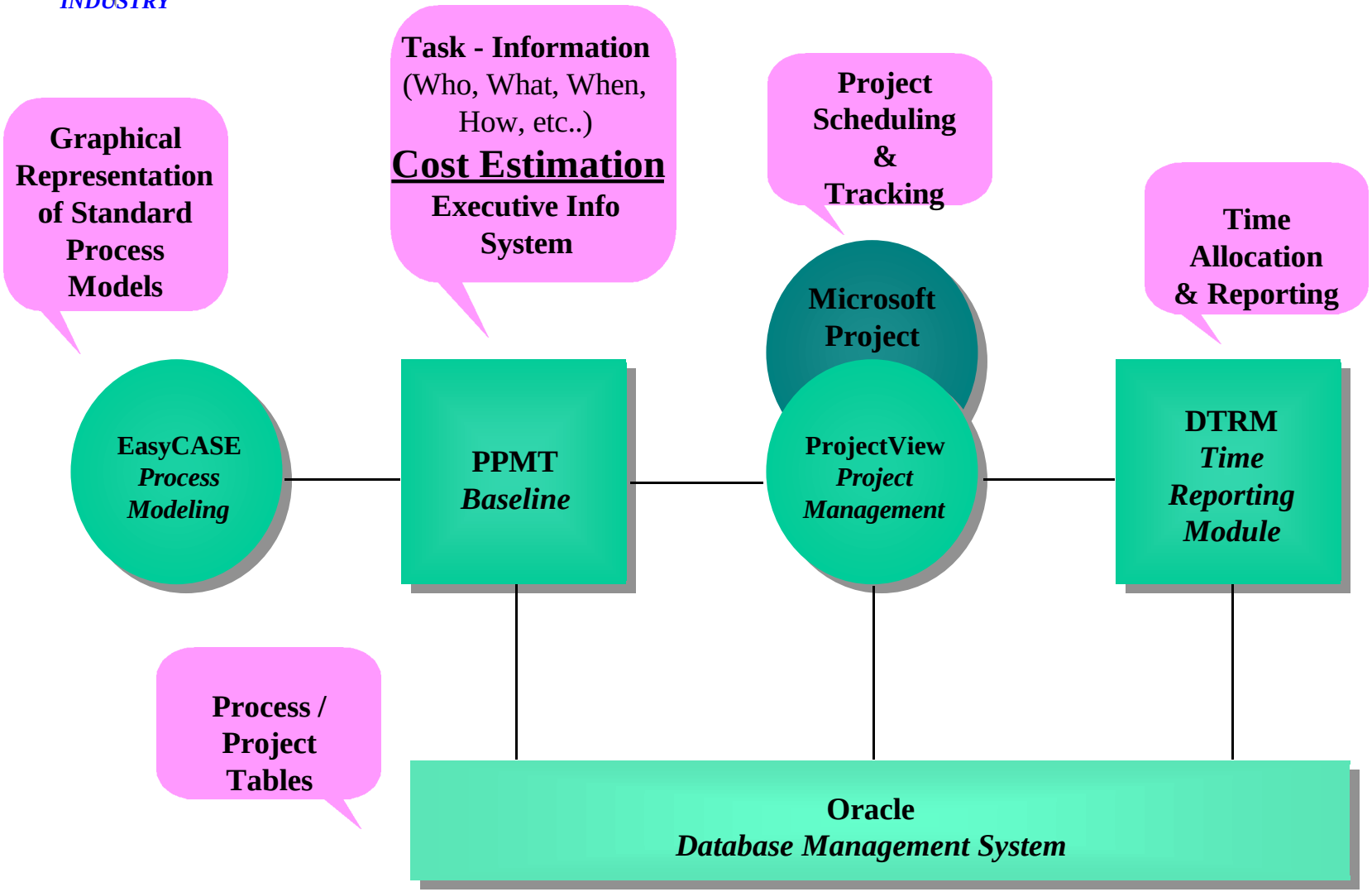
CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

Closer to the Mark

- Requirement Definition
- Requirements Control
- Process Models
- Mil-Std-498
- **PPMT**
  - **FP Analysis**
  - **COCOMO**

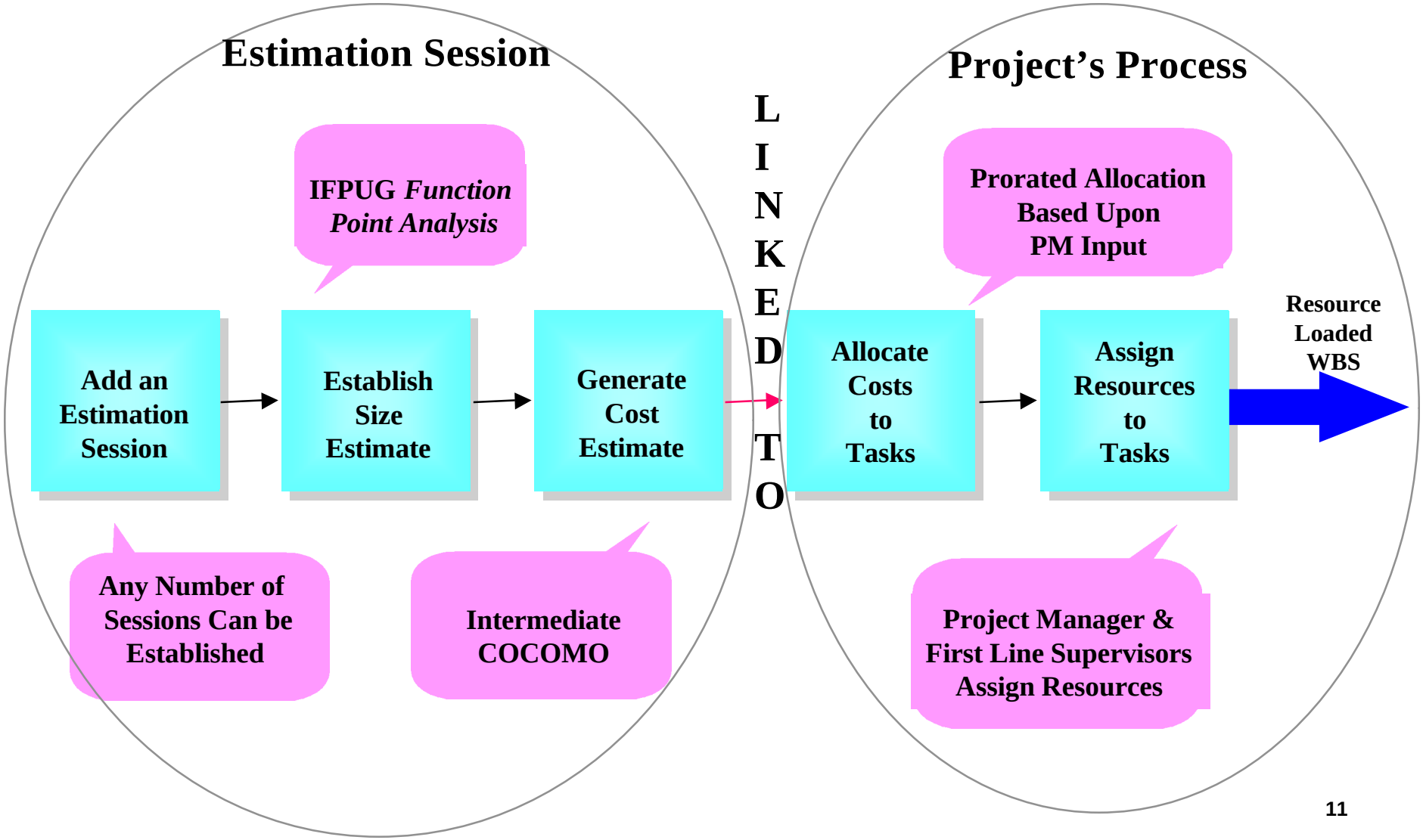
CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

Process & Project Management Toolset (PPMT)



CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

Size & Cost Estimation in PPMT



COCOMO
(COnstructive COst MOdel)

- B.W. Boehm, "*Software Engineering Economics*", 1981
- An accepted Industry Standard
- Based upon extensive research of previous information system projects
- Based upon software size (new and adapted) and 15 environmental cost drivers
- An Exponential Model - "Twice as big means more than twice as hard"

What is COCOMO?

The COCOMO model is really a rigorous extension of the "Analogy" method of estimation

- The model allows comparison of the current project to an extensive calibration database of completed projects
- It is as if we had completed these projects ourselves and had metrics for them
- It is repeatable - thus, it fosters consistency
- It is recognized world-wide as an industry standard mechanism for software estimation

CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

The Forms of COCOMO

- **BASIC:** Considers only size of the software system
- **INTERMEDIATE:** Based upon software size (measured in Delivered Source Instructions [DSI]) plus 15 general "cost drivers" that relate to the Product, Project, Personnel, and Environment complexity
- **DETAILED:** Intermediate COCOMO taken to the software component level. Complex and labor intensive.

DSDC has chosen Intermediate COCOMO as a good compromise between accuracy and effort and time required to obtain estimations

Intermediate COCOMO

Covers:

Development Phases

- Plans/Requirements Mgmt
- Product Design
- Programming
- Developmental Integration/Test

Activities

- Requirements Analysis
- Product Design
- Programming
- Developmental Test Planning
- Verification/Validation
- Project Office
- CM/QA
- Manuals/Documentation

Does not cover:

- Requirements Development
- OT & E

CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

Specifying the project to COCOMO

- Development Mode
- New Code Size
- Adapted Code Parameters
- Cost Drivers
- Productivity

COCOMO Development Modes

- ***Organic*** - Small applications developed in a stable, highly familiar, in-house environment using very small, cohesive teams, with flexible, non-demanding specifications
- ***Embedded*** - Tightly coupled complex of hardware, software, regulations, and procedures with very demanding specifications. Generally applicable to real-time systems such as avionics and communications
- ***Semi-Detached*** - Moderate to large applications and teams, strong specifications, and large user communities. Most DSDC projects fall into this mode.

Software Size - New Code

- Size of newly developed software, measured in Delivered Source Instructions (DSI)
- Sometimes referred to as “Lines of Code” (LOC) or Source Lines of Code (SLOC)
- Unfortunately, size is difficult to estimate directly from Requirements “when the paper is blank”
- Mechanisms do exist to assist in this estimate, however (Function Point Analysis)

The Software Estimation Paradox

All software estimation models exploit the correlation between software "size" and the resultant effort/schedule required to produce it.

- "Size" can be measured in just about any units, so long as one is consistent. (# listing pages, weight of listing, etc)
- The common size unit is Source Lines of Code (or equivalently, Delivered Source Instructions)
- Arguments rage over lines of code, but as long as one is consistent, they are pretty much academic

The Paradox: How do we estimate software size when it isn't built yet and all we have are functional requirements?

Enter Function Point Analysis

Function Point Analysis (FPA) attempts to solve this problem by modeling ultimate software size as a function of the characteristics and attributes of the requirements

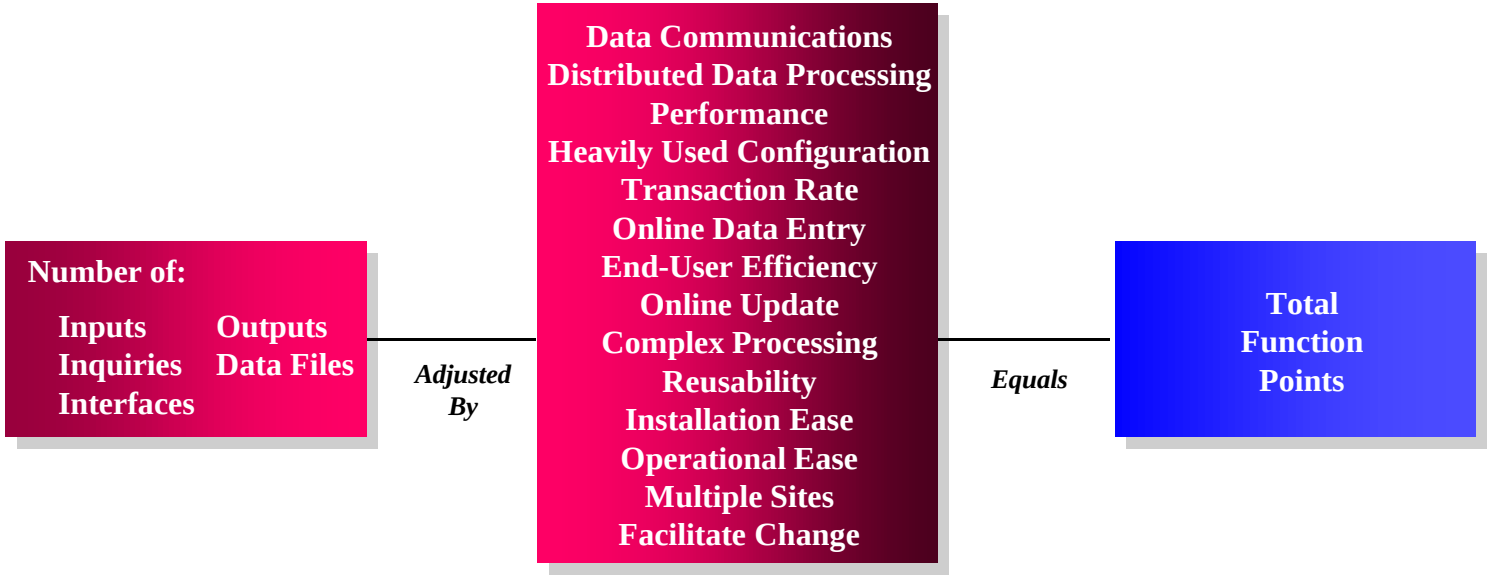
- Counts of inputs, outputs, files, interfaces, and inquiries, ranked by complexity (simple, average, complex)
- Rankings on 14 “Influence Factors” which describe the overall software product
- A pre-calibrated mapping of function points to DSI for a variety of programming languages in common usage (and some not-so-common ones!)

CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

Function Point Analysis

Definition:

An objective, quantitative measurement of the size and complexity of a system based upon the user point of view.



Total Function Points

X Language Conversion Factor

Estimated Lines of Code

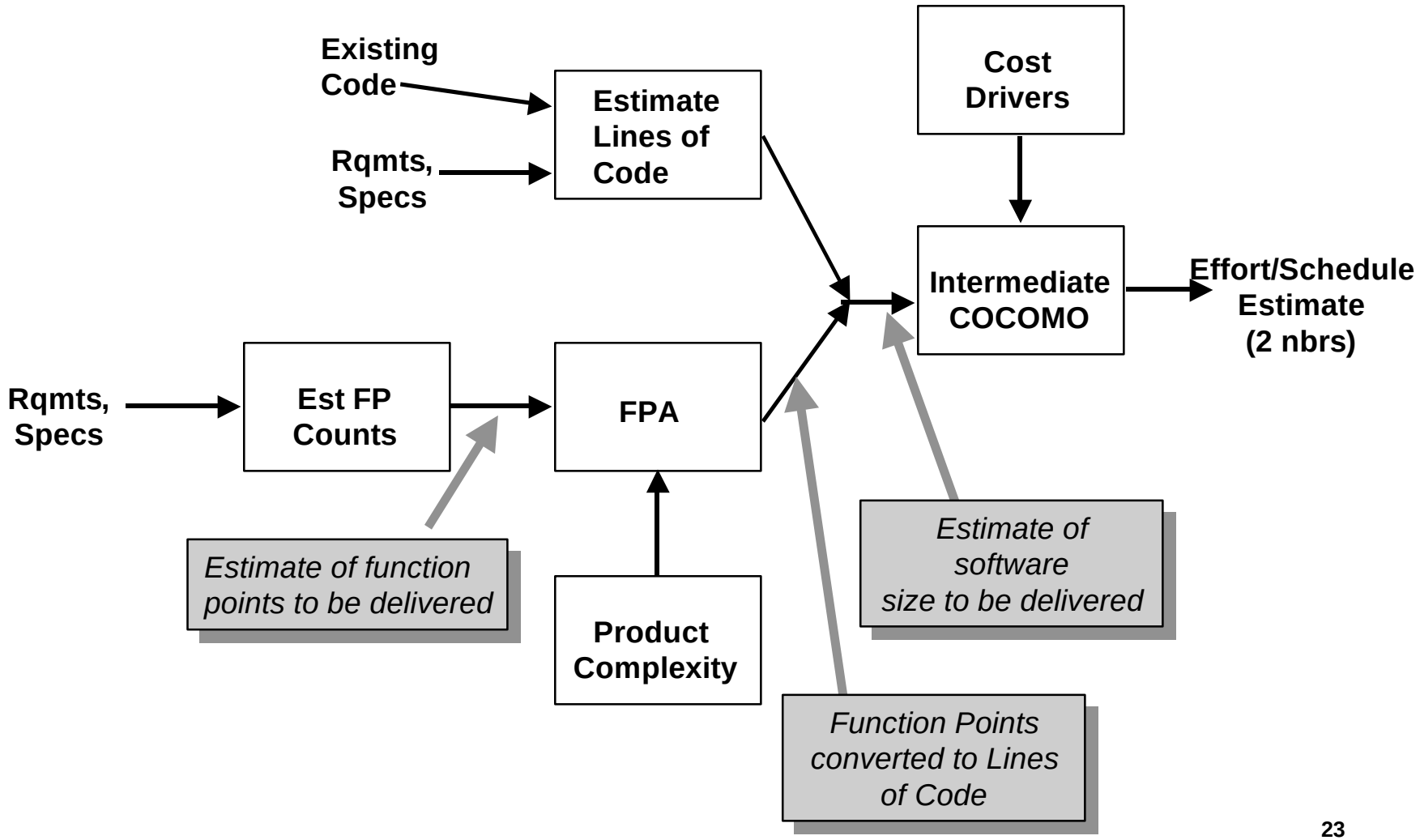
CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

Software Size - Adapted Software

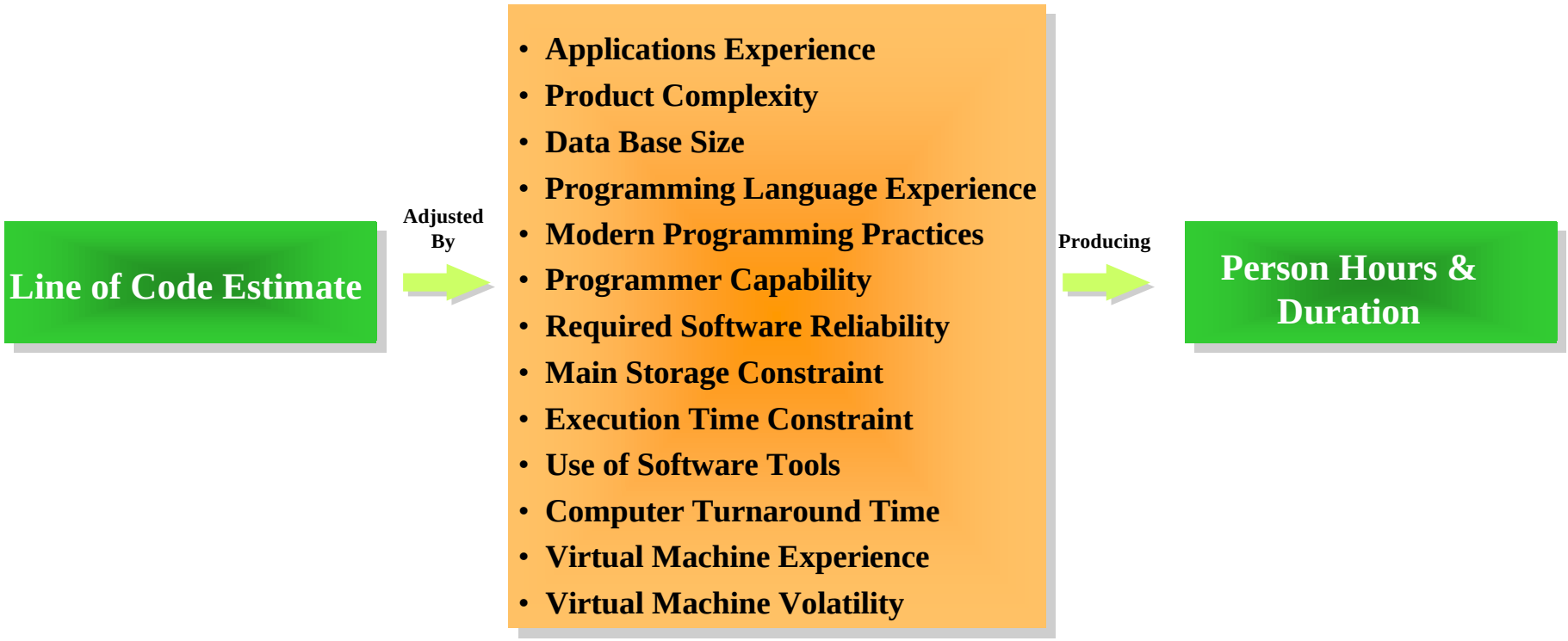
- Size of module(s) to be adapted
- Percent Design Modified
- Percent Code Modified
- Percent Full-scale Integration Test Required

New or Adapted Code

Software Estimation Process - COCOMO



COCOMO - Cost Drivers



The COCOMO Cost Drivers

The COCOMO Cost Drivers "tune" the model to the specific characteristics of the current job being estimated using four categories of factors:

- ***Product:*** Three factors describing the general nature of the software product to be produced
- ***Computer:*** Four factors describing the nature of the computer platform on which the product will be used and developed
- ***Personnel:*** Five factors relating the experience and capability of the development team to the work being undertaken
- ***Project:*** Three factors describing the nature of the project and the software engineering environment

COCOMO Product Cost Drivers

- **RELY: Required Software Reliability**

  Ranges from Very Low (failures are minor annoyances with work-arounds) to Extremely High (failure endangers human life and is unrecoverable)

- **DATA: Data Base Size**

  Ranges from Very Low (small database size [or size not relevant] to Very High (database size impacts project significantly)

- **CPLX: Product Complexity**

  Ranges from simple, straight-line processing to highly complex, multiple thread processes with complex algorithms

COCOMO Computer Cost Drivers

- **TIME: Execution Time Constraint**

  Ranges from none at all through extremely constrained (e.g. daily process must run in less than 43 minutes)

- **STOR: Main Storage Constraint**

  Memory or Disk Storage limitations, ranging from essentially none to very constrained

- **VIRT: Virtual Machine Volatility**

  Relative stability of the overall software development environment, including computer(s), networks, and tools

- **TURN: Computer Turnaround Time**

  Relative ability of developers to obtain required responses from the computer

COCOMO Personnel Cost Drivers

- **ACAP: Analyst Capability**

  Relative skill level of the System Analyst pool in relation to this task

- **AEXP: Application Experience**

  Degree of personnel pool experience with this application

- **PCAP: Programmer Capability**

  Relative skill level of the Programmer pool

- **VEXP: Virtual Machine Experience**

  Relative level of experience of the personnel pool with the development and execution environments

- **LEXP: Programming Language Experience**

  Relative experience level of the Programmer pool with the chosen language

COCOMO Project Cost Drivers

- **MODP: Modern Programming Practices**

  Relative degree to which modern software engineering practices will be employed in the development effort

- **TOOL: Use of Software Tools**

  Relative degree to which software engineering tools (e.g. CASE tools) will be employed in the development effort

- **SCED: Required Development Schedule**

  Degree to which the development schedule will be externally constrained (see next slide)

"What if I need it sooner"

Two well established software engineering principles:

- If you compress a software development schedule, the work will require more effort (staff hours)
- If you have "all the time in the world," you will take it

- Compression of the schedule leads to higher risks
- Compression to less than about 70% of nominal is generally unrealistic.
- One effective mechanism for schedule compression is to compress only the critical delivery tasks.

COCOMO does not lower the intrinsic risk of schedule compression, BUT it does make it quantitatively visible

COCOMO Output

COCOMO Produces:

- An estimate of Effort (staff-months) for the project
- An estimate of the nominal Schedule (months)
- A distribution by development phase and activity
- An estimated average staff size, based on productivity

COCOMO assumes "commercial documentation," and certain other characteristics about the software engineering methodologies. The raw results of the model can be scaled to accommodate variations from these assumptions

How Accurate is COCOMO for DSDC?

During model selection for PPMT, DSDC provided data for a number of completed projects. These data were used to create estimates from various models.

- Intermediate COCOMO estimates effort for Organic Mode projects with very low errors (average < 1%)
- Schedule estimates tends to be a little inflated, which is highly typical
- Large project data provided an extremely good fit, however number of projects was insufficient for statistical significance

The DSDC Linear Model
PM Guidance Enclosure 7.1

The DSDC Linear Model provides a simple, alternative estimation technique for cross-checking

- Based on Function Point Analysis
- Estimates the uncertainty as well as the value, and accommodates a variable amount of risk
- Assumes constant productivity regardless of project size

While linear models tend to be less accurate than exponential models (they do not account for diseconomies of scale), they <u>are</u> usually more stable

CUSTOMER
DSDC
PARTNERS
in Quality Solutions
INDUSTRY

Summary

- The DSDC Software Process Improvement effort to reach SEI CMM Levels 2 & 3 is providing a repeatable framework in which to perform software development and maintenance
- PPMT provides a centralized, shared database and process enactment tools to implement the repeatability
- Software effort/schedule estimation tools, imbedded in PPMT provide repeatable, consistent estimation
- The organization-wide shared database facilitates process improvement efforts, making estimations converge to DLA-DSDC reality